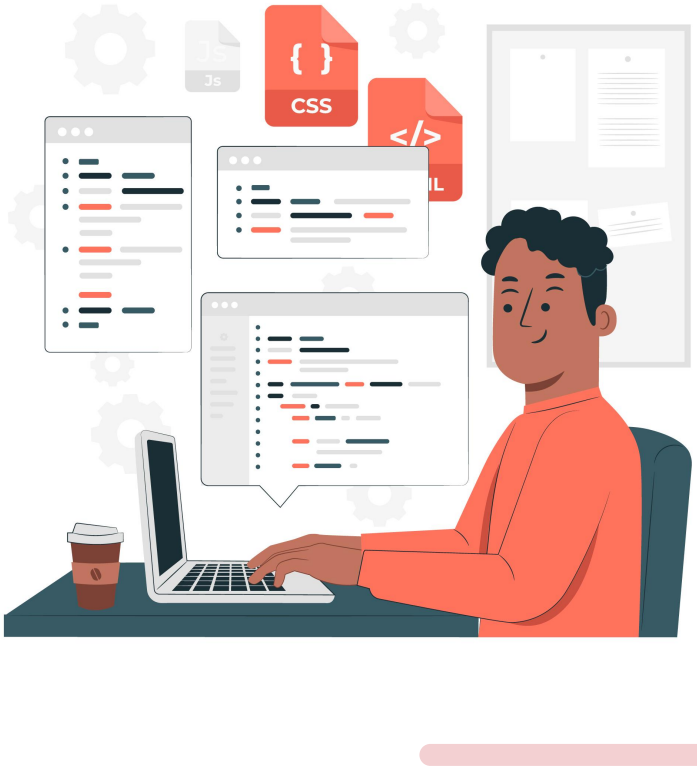
Pengenalan Express.js, Web Service API & Middleware



KEMNAKER

sanbercode

Outline

- Apa itu Express.js
- Instalasi Express.js
- Basic Routing Express.js
- Static File Express.js
- Web Service API
- Middleware di Express.js

KEMNAKER

Sanbercode

Apa itu Express.js

Apa itu Express.js

Express.js adalah framework web app untuk Node.js yang ditulis dengan bahasa pemrograman JavaScript. Framework open source ini dibuat oleh TJ Holowaychuk pada tahun 2010 lalu. Express.js adalah framework back end. Artinya, ia bertanggung jawab untuk mengatur fungsionalitas website, seperti pengelolaan routing dan session, permintaan HTTP, penanganan error, serta pertukaran data di server. Framework yang satu ini punya arsitektur MVC (Model View Controller). Dengan begitu, setiap data diolah pada Model, dihubungkan melalui Controller, lalu ditampilkan menjadi informasi melalui View.

Kelebihan dan Kekurangan Express.js

Kelebihan	Kekurangan
Bebas menentukan sendiri arsitektur dan struktur website yang akan dikembangkan	Diperlukan penelusuran kode lebih untuk membuat, mengelola, dan merawat arsitektur website yang sudah dibuat sebelumnya dengan express
Ukuran framework lebih kecil dan ringan, karena hanya berisi package inti	Terlalu banyak pilihan plugin dan library untuk digunakan, bisa membingungkan bagi sebagian orang
Kinerja website yang dihasilkan jadi lebih baik, mengingat ringannya framework	Pilihan metode pengembangan yang berbeda-beda, sehingga tidak ada standar baku

KEMNAKER

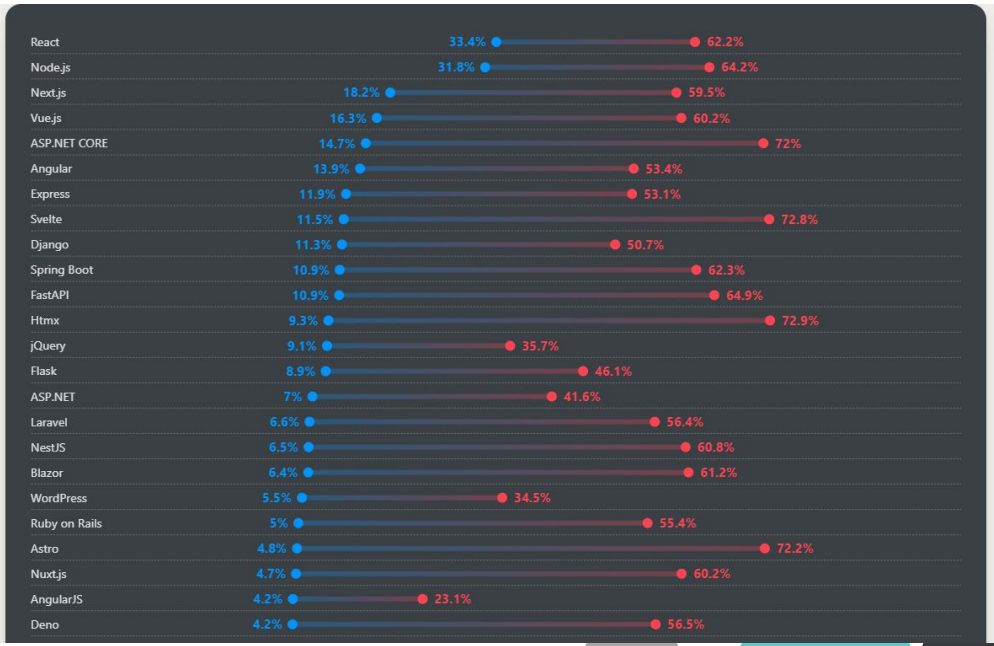
Hasil Survey Stackoverflow 2024

https://survey.stackoverflow.co/2024/technology#2-web-frameworks-and-technologies

Web frameworks and technologies

73% of developers that used it want to keep working with Svelte. Fun fact: Our team at Stack Overflow used Svelte for the first time in building our 2024 Developer Survey results site. We could go on and on about Svelte, listen to us do just that in a interview with one of our own.

Which **web frameworks and web technologies** have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)



KEMNAKER

Sanbercode

Installasi Express.js

Step Installasi

- https://expressjs.com/en/starter/installing.html
- https://expressjs.com/en/starter/hello-world.html

Step (1)

buatlah folder baru lalu masuk ke directory folder tersebut

```
$ mkdir myapp
$ cd myapp
```

setelah itu jalankan perintah

```
$ npm init
```

lalu enter semua dialog sampai berhasil membuat file package.json

lalu buatlah file .gitignore yang berisi

```
node_modules
.env
```

Step (2)

lalu jalankan perintal install express

npm install express
npm install nodemon

lalu di scripts tambahkan

```
  "scripts": {
    "test": "echo \"Error: no test
specified\" && exit 1",
    "start": "nodemon index.js"
  },
```

KEMNAKER

Sanbercode

Step (3)

buat file index.js lalu isi dengan kode dibawah ini:

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello Dunia!')
})

app.listen(port, () => {
  console.log(`App listening on port http://localhost:${port}`)
})
```

KEMNAKER

Sanbercode

Sesi Tanya Jawab

KEMNAKER

Sanbercode

Basic Routing Express.js

Basic Routing Express

cara menggunakan Routing pada express.js adalah seperti dibawah ini:

```
app.METHOD(PATH, HANDLER)
```

- **app merupakan instance dari express**
- **METHOD diisi dengan HTTP Method https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods**
- **PATH diisi dengan path dari route yang dibuat**
- **HANDLER diisi function yang harus memiliki dua parameter request dan response**

KEMNAKER

Sanbercode

Static File Express.js

Konfigurasi Dasar Static File di Express

cara menggunakan Static File di express.js adalah seperti dibawah ini:

```
const express = require('express');
const app = express();

app.use(express.static('public'));
```

ini berarti file yang ada dalam folder public itu bisa langsung diakses tapi bersifat statis, yang maksudnya file tersebut dikirim ke client tanpa diproses di server

KEMNAKER

Sanbercode

Sesi Tanya Jawab

KEMNAKER

Sanbercode

Web Service API

Apa itu Web Service API?

Web Service API adalah sebuah web yang menerima request dari client dan menghasilkan response, biasa berupa JSON/XML.

bentuk web Service API bisa berupa SOAP, REST API, dan lain-lain.

Apa itu REST API?

REST API (Representational State Transfer):

- **REST API adalah konsep arsitektur yang mendefinisikan aturan untukmembangun layanan web.**
- **Berfokus pada sumber daya (resources) dan tindakan (actions) yang diambil terhadap sumber daya tersebut.**
- **Menggunakan metode HTTP standar (GET, POST, PUT, DELETE) untuk berinteraksi dengan sumber daya.**
- **Tidak memiliki aturan khusus terkait dengan format data yang digunakan bisa menggunakan XML, JSON, atau format lainnya.**

Apa itu JSON?

JSON (JavaScript Object Notation) adalah text format yang digunakan untuk menyimpan data, bentuk umumnya seperti javascript object.

Contoh format JSON:
{"nama": "John", "usia": 30}

Contoh Data Dummy untuk Demo

```
let movies = [
  {id: 1, title: "Spider-Man", year: 2002},
  {id: 2, title: "John Wick", year: 2014},
  {id: 3, title: "The Avengers", year: 2012},
  {id: 4, title: "Logan", year: 2017},
]
```

Demo membuat REST API

KEMNAKER

Sanbercode

Sesi Tanya Jawab

KEMNAKER

Sanbercode

Middleware di Express

Middleware

Middleware adalah satu satu mekanisme keamanan website. Middleware berjalan sebelum fungsi utama di proses.

Sebagai contoh sederhana, misalnya terdapat routing untuk mengarahkan ke dalam fungsi, fungsi tersebut digunakan untuk melihat data. Agar fungsi tersebut tidak dapat di akses orang maka harus melewati middleware, middleware berperan untuk menghentikan atau meneruskan proses.

KEMNAKER

Sanbercode

Membuat Middleware

```
const express = require('express')
const app = express()
const port = 8080

app.use(express.json())

const logMiddleware = (req, res, next) =>{
  console.log("Ini dari middleware Log....")
  next()
}

const getText = (req,res)=>{
  res.status(200).json({text: "this is text"})
}

app.get('/text', logMiddleware, getText)

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Middleware dengan memeriksa parameter URL

```
const express = require('express')
const app = express()
const port = 8080

app.use(express.json())

const tokenCheck = (req, res, next) =>{
  let {token} = req.query

  if (token !== '12345'){
    res.status(400).send('Token tidak tersedia atau salah')
  }else{
    next()
  }
}

const getMovie= (req,res)=>{
  res.status(200).send('<h1>Anda Berhasil Mengakses Fungsi GetMovie() </h1>')
}

app.get('/', tokenCheck, getMovie)

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

KEMNAKER

Sanbercode

Middleware Basic Auth

```
const basicAuth = (req, res, next) => {
  var authheader = req.headers.authorization;

  if (!authheader) {
    var err = "You are not authenticated!"
    res.setHeader('WWW-Authenticate', 'Basic');
    err.status = 401;
    return next(err)
  }

  var auth = new Buffer.from(authheader.split(' ')[1], 'base64').toString().split(':');
  var user = auth[0];
  var pass = auth[1];

  if (user == 'admin' && pass == 'admin') {
    // If Authorized user
    next();
  } else {
    var err = "You are not authenticated!"
    res.setHeader('WWW-Authenticate', 'Basic');
    err.status = 401;
    return next(err);
  }
}
```

KEMNAKER

sanbercode

Kesimpulan

- Apa itu Express.js
- Instalasi Express.js
- Basic Routing Express.js
- Static File Express.js
- Web Service API
- Middleware di Express.js

KEMNAKER

Sanbercode

Terima Kasih